**grafik, siyah beyaz, tasarım içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**

**Bursa Teknik Üniversitesi**

**Bilgisayar Ağları**

**Low-Level IP İşleme ve Ağ Performans Analizi ile Gelişmiş Güvenli Dosya Transfer Sistemi**

*Proje Ara Raporu*

**Ad:** Eren

**Soyad:** KÖSE

**Öğrenci Numarası:** 22360859075

**Introduction - Tanıtım**

* Bu projede, şifreleme, kimlik doğrulama ve bütünlük kontrolü sağlayan gelişmiş bir dosya transfer sistemi geliştirilmekte. Sistem, IP başlıklarının düşük seviyeli işlenmesi (TTL, bayraklar, parçalara ayırma, checksum) yoluyla ağ protokollerine derinlemesine bir bakış sunmayı hedeflemektedir. Aynı zamanda, ağ performansı (gecikme, bant genişliği, paket kaybı) çeşitli senaryolar altında analiz edilmekte ve güvenlik testleri (MITM saldırıları, paket enjeksiyonu) gerçekleştirilmektedir. Python (Scapy) tabanlı bu sistem, güvenli iletişim ve performans analizi konuları deneyimi amaçlamaktadır.

**Technical Details – Teknik Detaylar**

* Yazılım Dili 🡪 Python
* Şifreleme Kütüphaneleri 🡪 cryptography
* Diğer Kütüphaneler 🡪 Scapy, socket, secrets, subprocess
* Ağ Analiz Araçları 🡪 ping, iperf3, Wireshark
* metin, ekran görüntüsü, yazı tipi, tasarım içeren bir resim

  Yapay zeka tarafından oluşturulan içerik yanlış olabilir.Paket Manipülasyonu 🡪 scapy
* Dosya Sistemi 🡪
* Sunucuyu başlatmak için “python main.py server” komutunu terminale giriyoruz. Bu bizim TCP[1] sunucumuzu başlatıyor, gelen bağlantılara RSA[2] public anahtarını gönderiyor, AES[3] anahtarını çözüyor.

metin, ekran görüntüsü, iş kartı, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

* + start\_server fonksiyonu sunucuyu başlatır, bağlantı bekler, istemciden veri alır ve dosyayı deşifre ederek kaydeder. TCP socket açılır ve 9999 numaralı port üzerinden dinlenir, 0.0.0.0 ile her IP’yi kabul eder. Çok sayıda bağlantı durumu yaşanmaması amacıyla da 5 adet IP dinlemesine izin vermesi sağlanır. Ayrıca diğer anlatılacak olan fonksiyonlar da bu fonksiyonun içinde çağırılmaktadır.
    - metin, ekran görüntüsü, multimedya, yazı tipi içeren bir resim

      Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Bu kısım start\_server fonksiyonu içinde RSA public keyini client tarafına gönderdiğimiz kısımdır.

* + - metin, iş kartı, ekran görüntüsü, yazı tipi içeren bir resim

      Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, elektronik donanım, ekran görüntüsü, ekran, görüntüleme içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.Bu kısım start\_server fonksiyonu içinde şifrelenmiş olan AES anahtarının şifresinin çözüldüğü kısımdır.

Bu kısım start\_server fonksiyonu içinde şifrelenmiş dosyanın clienttan alınıp şifresinin çözüldüğü kısımdır. Sonrasında ise alınan dosya kaydedilir ve o clienta olan erişim kapatılır.

metin, ekran görüntüsü, yazı tipi, yazılım içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

* + generate\_keys fonksiyonu sunucunun RSA anahtar çifti üretmesini sağlar. Public key sunucuya gönderilir, private key ise sunucuda gizli tutulur. rsa fonksiyonu ile yapılır.

metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

* + decrypt\_aes\_key\_with\_rsa fonksiyonu client tarafından gelen AES anahtarını, RSA-OAEP (Optimal Asymetric Encryption Padding) kullanılarak şifre çözülür. Koruma ise SHA-256 hash algoritması kullanılarak sağlanır.
  + Sunucu başlatıldıktan sonra şekildeki ekran çıktısını verir ve 9999 portunu dinlemeye başlar.
* Başka bir terminale gelip “python main.py client --file example.txt --ip <SERVER\_IP>” komutunu girdiğimizde ise client tarafından IP’si girilen servera dosya aktarımı yapabiliyoruz. Burada sunucuda RSA public anahtarı alınıyor, AES anahtarı oluşturulur ve şifreleme yapılır, SHA256 hash oluşturulur ve dosya paylaşılır.

metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

* + encrypt\_file\_with\_aes fonksiyonu ile gönderilecek veri AES algoritması ile şifrelenir. Rastgele 16 bit uzunluğunda bir vektör üretilir. Dosya verisine gerekli padding yapılarak bloklara tamamlanılır. Bu sayede veri şifrelenmiş olur.

metin, ekran görüntüsü, yazı tipi, multimedya içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

* + calculate\_sha256 fonksiyonu ile gönderilen dosyanın büyüklüğünü doğrulama amaçlı bir SHA-256 hash değeri üretilir.

metin, iş kartı, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

* + start\_client fonksiyonu ile ise TCP sunucuya bağlanılır, RSA public key alınır, rastgele bir AES anahtarı üretilir, AES anahtarı RSA ile şifrelenip sunucuya gönderilir, dosya da AES ile şifrelenir ve dosya da sunucuya gönderilir.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

start\_client fonksiyonu içindeki RSA public key alma kısmıdır. Bu sayede AES keyi şifrelenebilir.

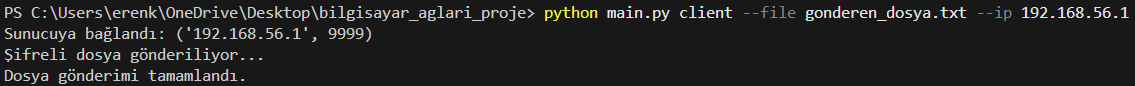
metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

start\_client fonksiyonu içindeki AES anahtarı oluşturma ve bu anahtarı RSA ile şifreleme kısmıdır.

metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

start\_client fonksiyonu içindeki AES anahtarını servera gönderme, dosyayı AES anahtarı ile şifreleme, dosya hashini hesaplama ve dosyayı gönderip clientı kapatma kısımlarıdır.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir. Client kodu çalıştırıldığında ekran çıktısı client kısmında bu şekildedir.

Client kodu çalıştırıldığında ekran çıktısı server kısmında bu şekildedir.

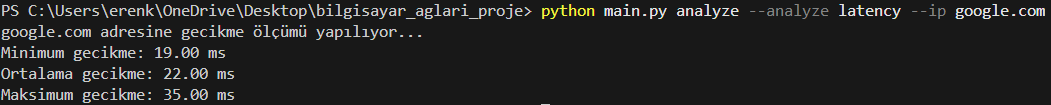
* Latency ölçümü için “python main.py analyze --analyze latency --ip 8.8.8.8” komutunu terminale girdiğimizde mesela Google.com’a olan gecikmeyi ölçebilmekteyiz. IP adresini değiştirip istediğimiz servera olan latencymizi ölçebiliriz.

metin, elektronik donanım, ekran, görüntüleme, ekran görüntüsü içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.metin, elektronik donanım, ekran, görüntüleme, ekran görüntüsü içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

* + measure\_latency fonksiyonu ile belirtilen hedef IP’ye ping komutu ile ping atılır ve gecikme ölçülmeye çalışılır. Ping komutu sonrası alınan cevaptan ihtiyacımız olan kısımları filtreliyoruz. Sonrasında çok küçük olan gecikmeleri belirli bir değere sabitliyoruz. Ping komutunun herhangi bir şekilde işlememesi sonucunu ise try-except durumu ile kontrol ediyoruz.



Bu şekilde örneğin Google.com sitesine olan latencyi ölçebiliriz. Ekran çıktısı bu şekildedir.

* metin, ekran görüntüsü, bilgisayar, ekran, görüntüleme içeren bir resim

  Yapay zeka tarafından oluşturulan içerik yanlış olabilir.Bandwith ölçümü için öncelikle terminale gelip “iperf -s” komutu ile server olarak iperf’i[3] çalıştırıyoruz. Sonrasında ise farklı bir terminalde ise client olarak “python main.py analyze --analyze bandwidth --ip <SERVER\_IP>” komutunu çalıştırıyoruz ve bu sayede ağın bant genişliğini ölçebiliyoruz.
  + run\_iperf\_client fonksiyonu ile hedef IP üzerinde çalışan bir iPerf sunucusuna bağlanarak bant genişliği ölçebiliriz. iPerf üzerinden gelen çıktıyı algılayıp istenen sonuç yine filtrelenir, kullanıcıya geri dönüş sağlanır.
  + 

iPerf sunucusuna bağlanıp bant genişliği ölçümü sonrası elde edilen ekran çıktısı bu şekildedir.

* Paket kaybı ve ağ güvenlik analizi kısımları şu anda yapım aşamasındadır. Teknolojileri ve nasıl yapabileceğimi araştırmaya devam ediyorum.

**Limitations and Improvements – Kısıtlamalar ve Geliştirmeler**

* Şu anda geliştirmemiş olmama rağmen bu proje üzerine bir Graphical User Interface (GUI) tasarlayıp kodlamayı da planlamaktayım.
* Ağ analizinde eksik paket yollama denemesinde tc kullanmayı deneyecektim ancak yaptığım araştırmalar sonucunda tc’nin yalnızca Linux’ta kullanıldığını öğrendim. Bu yüzden onun yerine Clumsy[5] kullanmayı planlıyorum.
* Wireshark[6] üzerinden iletimleri nasıl izleyeceğimi biraz çözdüm fakat tam anlayamadım, bundan sonraki kısımlarda oraya odaklanıp anlamaya çalışacağım.

**Conclusion - Sonuç**

Bu proje, güvenli dosya iletimi, düşük seviyeli IP paket işleme ve ağ performansı analizi konularını birleştiren, ileri düzeyde bir bilgisayar ağları uygulamasıdır. Projenin temel amacı, ağ üzerinden dosya gönderimini şifreli ve bütünlüğü korunmuş bir biçimde gerçekleştirmek, aynı zamanda bu aktarımı düşük seviyeli IP düzeyinde manuel olarak kontrol etmektir. Dosyalar AES algoritması kullanılarak simetrik olarak şifrelenmiş ve bu anahtar RSA ile asimetrik olarak güvenli biçimde sunucuya aktarılmıştır. Şifreleme işlemleri sırasında cryptography ve hashlib kütüphaneleri kullanılarak hem gizlilik (confidentiality) hem de bütünlük (integrity) sağlanmıştır. Gönderilen dosyanın SHA-256 özeti, alıcıda doğrulama için kullanılmıştır.

Dosya aktarımı sırasında düşük seviyeli IP paketleri Scapy kütüphanesi kullanılarak doğrudan yapılandırılmıştır. IP başlıklarında TTL, fragment offset, flags ve checksum alanları manuel olarak değiştirilmiş ve büyük veriler paketlere bölünerek fragmentasyon gerçekleştirilmiştir. Bu sayede, hem IP düzeyinde veri iletimi hem de yeniden birleştirme (reassembly) süreçleri uygulamalı olarak test edilmiştir. Ek olarak checksum hesaplamaları, olası iletim hatalarını tespit etmek üzere özel olarak kodlanmıştır.

Ağ performans analizi modülünde, gecikme ölçümü için sistem bağımlı olarak ping komutu kullanılmış, bant genişliği ölçümleri ise iperf aracı ile yapılmıştır. Clumsy adlı harici bir araç ile paket kaybı ve ağ gecikmesi simüle edilmesi planlanmaktadır. Böylece sistem, farklı ağ koşulları (Wi-Fi, Ethernet, sanal ağlar vb.) altında test edilerek analiz edebilecektir.

Projenin güvenlik analiz bileşeninde ise, ağ trafiği dinlenecek ve analiz edilecektir. Scapy ile gerçek zamanlı olarak filtreli paket yakalama işlemleri yapılacak, şifreli veri içeren paketler entropi analizine tabi tutulacaktır. Bu analizle, yakalanan verinin rastgeleliğe dayalı olarak şifreli olup olmadığı değerlendirilecektir. Ayrıca Man-in-the-Middle (MITM) saldırısı bir ARP spoofing simülasyonu ile gerçekleştirilecek ve sahte HTTP istekleri gönderilerek paket enjeksiyonu yapılacaktır. Bu işlemler sonucunda, sistemin veri güvenliği açıklarına karşı dayanıklılığı da test edilecektir.

Genel olarak, bu proje şifreleme tekniklerinden düşük seviyeli IP işleme yapılarına, ağ performans analizinden güvenlik testlerine kadar pek çok ağ teknolojisini bütüncül bir yapıda bir araya getirecektir. Hem istemci hem de sunucu tarafında platforma özel koşullar gözetilerek uyumlu çalışma sağlanacak, analiz ve simülasyon modülleri sayesinde kullanıcıya çeşitli test senaryoları sunulacaktır. Proje, ağ katmanlarının yapısını, güvenlik protokollerinin uygulamasını ve performans testlerini derinlemesine öğrenmek için kapsamlı ve teknik açıdan güçlü bir örnek teşkil etmektedir.

**References - Kaynakça**

* [1] Comer, D. E. (2018). *Internetworking with TCP/IP: Principles, Protocols, and Architecture* (6th ed.). Pearson.
* [2] Milanov, E. (2009). The RSA algorithm. *RSA laboratories*, *1*(11).
* [3] Daemen, J., & Rijmen, V. (2002). *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer Science & Business Media.
* [4] Taso, K., & Tirumala, A. (2005). *iPerf: Bandwidth Measurement Tool*. <https://iperf.fr>
* [5] Jagt. (2016). *Clumsy* [Computer software]. GitHub. <https://github.com/jagt/clumsy>
* [6] The Wireshark Team. (n.d.). *Wireshark user documentation*. Wireshark. <https://www.wireshark.org/docs/>